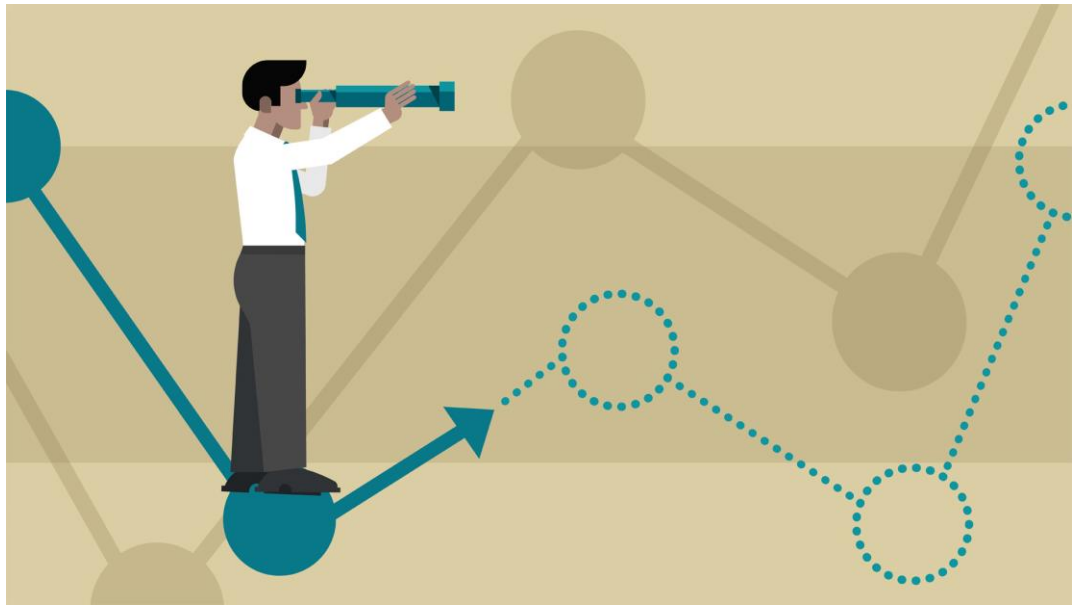
Forecasting

Business Intelligence per i Servizi Finanziari 2023-2024

Antonio Candelieri

Forecasting

- In the analysis of financial time series we are interested in designing models for **predicting future values**.



Linearity Assumption!

- *If the variables X_{t+h} , X_t , …, X_{t-p} have a **normal distribution**, i.e., the process $\{X_t\}$ is **Gaussian**, then*
  - $E(X_{t+h} | X_t , X_{t-1} , \dots , X_{t-p}) = a_0 X_t + a_1 X_{t-1} + \dots + a_p X_{t-p}$
- *where a_0 , …, a_p are real numbers*

- The problem of building a best predictor for a Gaussian process is solved by **forming a linear regression**.

- For any other process, **not necessarily normally distributed**, it is **still desirable** to design **a predictor as a linear combination of its past history**, even if this does not coincide with the conditional expectation of the process given its past history.

Linearity assumption

- In this case we want a linear function L of $\{X_t , X_{t-1} , \dots , X_{t-p}\}$ that minimizes the mean square error $E(X_{t+h} - L)^2$; that is, we want to find coefficients a_0 , a_1, …, a_p to form

$$L(X_{t+h}) = \sum_{j=0}^{p} \alpha_j X_{t-j}$$

- and such that their values minimize

$$F(\alpha_0, \dots, \alpha_p) = E\left(X_{t+h} - \sum_{j=0}^{p} \alpha_j X_{t-j}\right)^2$$

- This F is a quadratic function bounded below by 0, and hence there exists values of $(a_0 , \dots , a_p)$ that minimizes F, and this minimum satisfies

$$\frac{\partial F(\alpha_0, \dots, \alpha_p)}{\partial \alpha_j} = 0, j = 0, \dots, p.$$

Time Series Models in Finance

- The general paradigm for modeling a time series $\{X_t\}$ is to consider each term composed of a **deterministic component** H_t and a **random noise component** Y_t , so that

$$X_t = H_t + Y_t$$

- Then one can readily estimate the deterministic component H_t through some algebraic manipulations, and we are left with the difficult task of approximating the values of Y_t , that is the random component. The basic structure of a time series model for Y_t has the form

$$Y_t = E(Y_t \mid F_{t-1}) + a_t$$

where F_{t-1} represents the information set available a time $t-1$, $E(Y_t|F_{t-1})$ is the conditional mean, a_t is the stochastic shock (or innovation) and assumed to have zero conditional mean, and hence the conditional variance:

$$Var(Y_t|F_{t-1}) = E(a_t^2|F_{t-1}) = \sigma_t^2.$$

Time Series Models in Finance

- According to the form of $E(Y_t|F_{t-1})$ and $Var(Y_t|F_{t-1})$ as functions of F_{t-1}, we have models of different nature for Y_t. For example, if we fix the conditional variance to a constant value, $Var(Y_t|F_{t-1}) = \sigma^2$, and F_{t-1} is a finitely long recent past of Y_t, that is, $F_{t-1} = \{Y_{t-1}, \ldots, Y_{t-p}\}$, we obtain a linear model of the ***Autoregressive Moving Average*** (**ARMA**) type.

- If we leave both the conditional mean and conditional variance to vary as function of F_{t-1}, we obtain a nonlinear model of the *Autoregressive Conditional Heteroscedastic* (ARCH) type.

- Other **nonlinear** models can be obtained by using some nonlinear function $g()$, that is $Y_t = g(E(Y_t|F_{t-1}) + a_t)$, and assuming conditional mean or conditional variance constant or not. **Machine Learning** – such as **neural networks** and **support vector machines** – are widely adopted for this purpose.

Trend and Seasonality

- Taking into account the previous equation

$$X_t = H_t + Y_t$$

- The deterministic component H_t comprises the ***trend*** and the ***seasonality*** that might be present individually or jointly in the series.

- From a plot of the series one can check for the existence of a trend (observing if several mean values of the series go up or down in general), and check for a possible seasonal component (observing if some values repeat periodically).

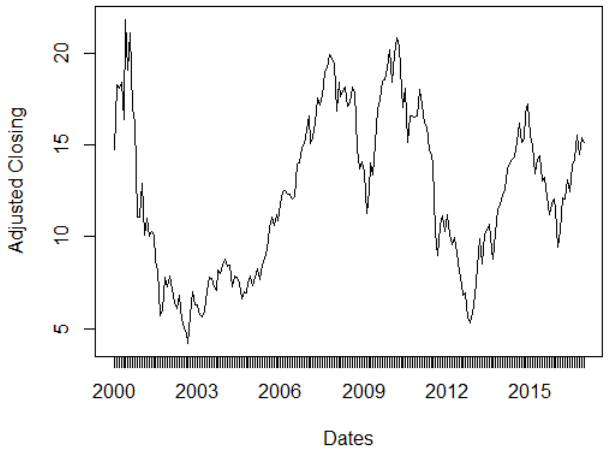
Time Series Models in Finance

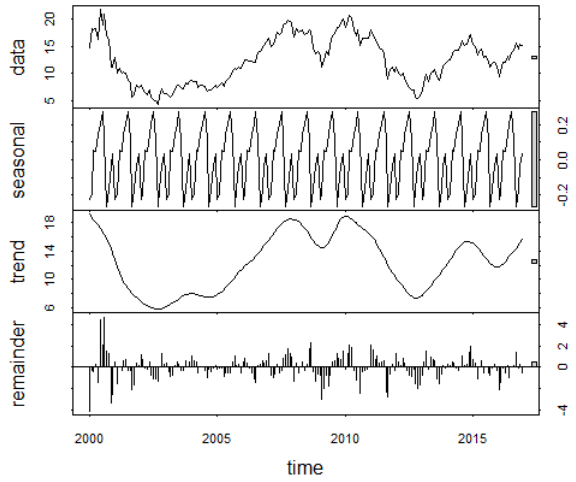
- If one perceives a trend or a seasonal component in the data then fits a model with trend or seasonality. We have then the ***classical decomposition model***

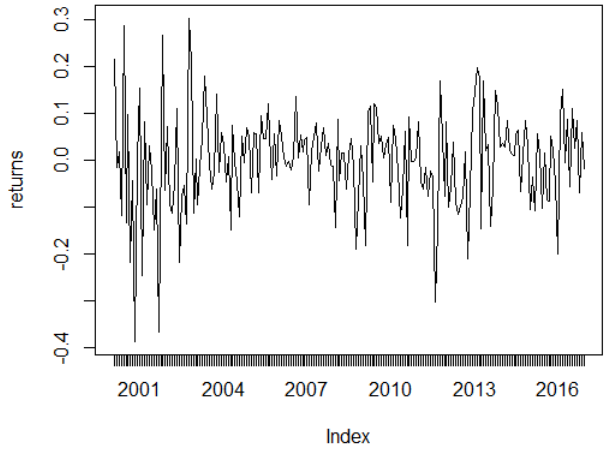
$$X_t = m_t + s_t + Y_t$$

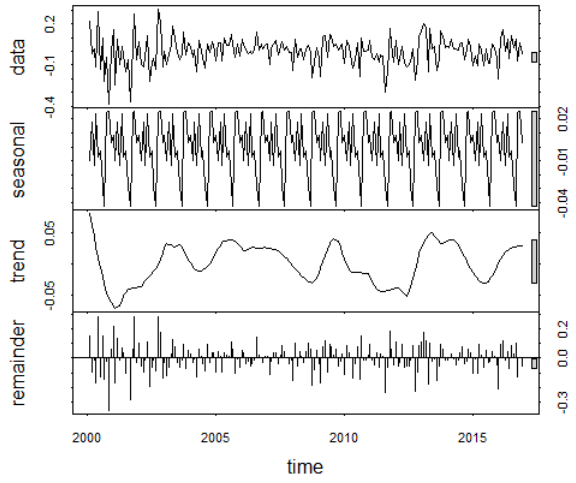
- where m_t is the trend component, s_t is the seasonal component, and Y_t is the random component.

  - the **trend component** is a slowly changing function (i.e., a polynomial in t: $m_t = a_0 + a_1 t + a_2 t^2 + \ldots + a_k t^k$ for some k), that can be approximated with least square regression;
  - the **seasonal component** is a periodic function, with a certain period d (i.e., $s_{t-d} = s_t$), that can be approximated with harmonic regression


Adjusted Closing
5
10
15
20
2000
2003
2006
2009
2012
2015
Dates


data
seasonal
trend
remainder
2000
2005
2010
2015
time


returns
-0.4
-0.2
0.0
0.1
0.2
0.3
2001
2004
2007
2010
2013
2016
Index


data
seasonal
trend
remainder
2000
2005
2010
2015
time

Estimating the random component

- Once the deterministic components are estimated, their estimations $H_t = m_t + s_t$ *can be* removed from the data to leave only the (sample) noise:
  - $Y_t = X_t - H_t$

- Alternatively, the trend or seasonality can be removed directly (without estimating them) by applying appropriate differencing operations to the original series X_t . For example:
  - a linear trend is removed by taking first differences, $X_t - X_{t-1}$;
  - a quadratic trend is removed taking second order differences, $X_t - 2X_{t-1} + X_{t-2}$, which is the same as taking first differences to $X_t - X_{t-1}$;
  - and so on.

Linear Processes and ARMA models

- A time series $\{X_t\}$ is a *linear process* if it has the form

$$X_t = \sum_{k=-\infty}^{\infty} \psi_k W_{t-k}$$

- for all t, where $\{\Psi_k\}$ is a sequence of constants with

$$\sum_{k=-\infty}^{\infty} |\psi_k| < \infty$$

- and $\{W_t\}$ is a (weak) white noise (uncorrelated random variables) with zero mean and variance σ^2; in symbols $\{W_t\} \approx WN(0, \sigma^2)$.
- The condition $\sum_{k=-\infty}^{\infty} |\psi_k| < \infty$ is to ensure that the series in the equation converges.

The classes of ARMA processes

- Consider a weak white noise, $\{W_t\} \approx WN(0, \sigma^2)$, and let integers $p \geq 1$ and $q \geq 1$. A time series $\{X_t\}$ is

AR(p) (autoregressive of order p) if

$$X_t = W_t + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p}$$

MA(q) (moving average of order q) if

$$X_t = W_t + \theta_1 W_{t-1} + \cdots + \theta_q W_{t-q}$$

ARMA(p, q) (autoregressive and moving average of order p, q) if

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + W_t + \theta_1 W_{t-1} + \cdots + \theta_q W_{t-q}$$

where $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ are real numbers, in all three equations.

Nonlinear (Semiparametric) Models

- **Artificial Neural Networks** (ANN or NNet)
- **Support Vector Machine** (SVM)

Semiparametric means that although the NNet and SVM – just as other more "traditional" methods – try to relate a set of input variables and weights to a set of one or more output variables, they differ from other methods in that their parameters are not limited in number, nor do they have to be all known or computed for the method to give an approximate solution.

In this regard is that these are known as **Machine Learning** approaches, since they fit in the concept of being algorithms that improve their performance at some task through experience.

Look at this small set of data…

	A	B	C
1	2.6	5.6	Y
2	2.7	3.6	Y
3	2.9	4.9	Y
4	2.6	2.7	Y
5	2.9	5.9	Y
6	2.3	5.5	Y
7	2.1	5.3	Y
8	2.4	3.4	Y
9	2.2	4.4	Y
10	2.7	3.7	Y
11	5.2	5.7	N
12	5.7	5.6	N
13	5.2	3.5	N
14	4.2	4.1	N
15	4.6	2.4	N
16	4.9	5.1	N
17	6.0	2.5	N
18	5.2	5.1	N
19	5.3	2.7	N
20	4.4	3.1	N

- Is there any relation linking the values of **A** and/or **B** to the value of **C**?
- In other terms: can you predict **C** depending on **A** and/or **B**?

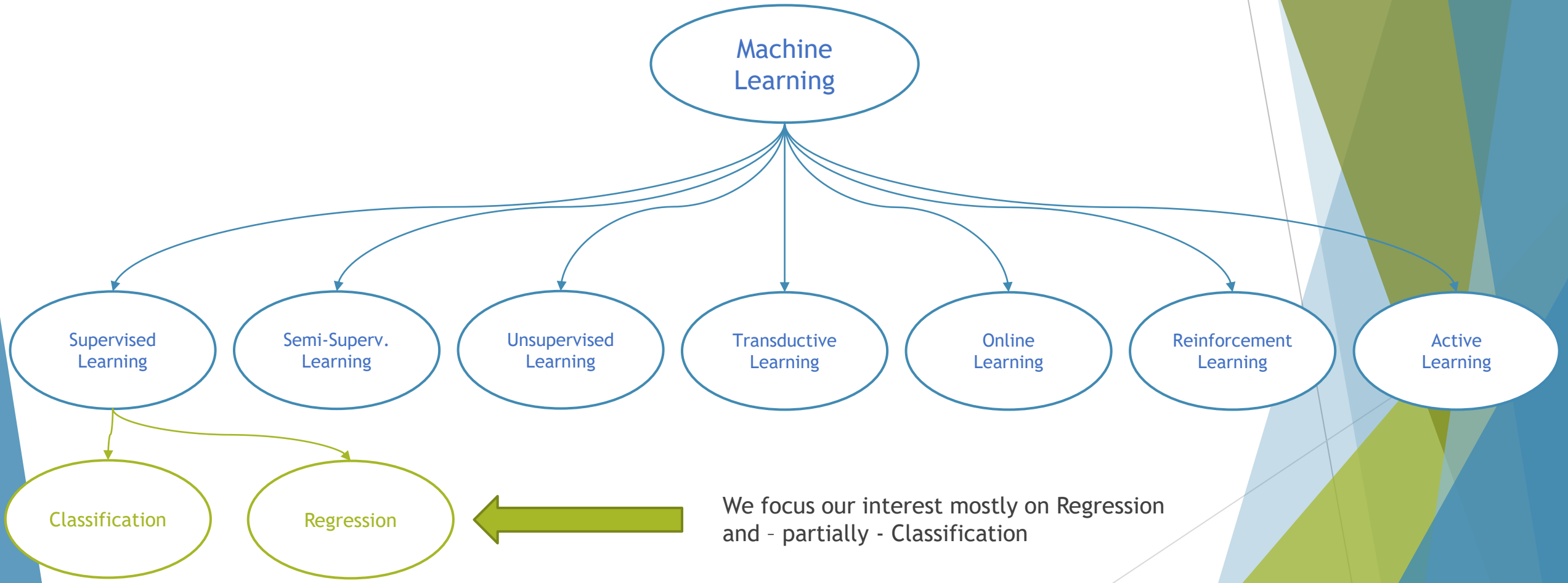
- The answer is easy… YES!
- For instance: **if** A <= 2.9 **then** C="Y" **else** C="N"
- But also… **if** A>=4.2 **then** C="N" **else** C="Y"

- As well as many others equivalent "rules" involving A…

- Basically you have "learned" from data!

- But what's about learning from millions/billions of rows and thousand and more of columns?!

Machine Learning in a nutshell



We focus our interest mostly on Regression and – partially - Classification

Learning Scenarios

- **Supervised Learning** – the learner receives a set of labeled examples (aka instances) as training data and makes predictions for unseen examples. This is the most common scenario associated with classification, regression, and ranking problems.

- **Unsupervised Learning** – the learner exclusively receives unlabeled training data, and makes predictions for unseen data. Since in general no laballed example is available in this setting, it can be difficult to quantitatively evaluate the performance of a learner. Clustering, anomaly detection, and dimensionality reduction are typical problems for this scenario.

- **Semi-supervised Learning** – the learner receives a set of both labeled and unlabeled training data, and makes predictions for unseen data. This scenario is common in settings where unlabeled data is easily accessible but labels are expensive to obtain. The idea is that the distribution of unlabeled data can help the learner achieve a better performance than in the supervised setting.

- **Transductive Learning** – as in the semi-supervised scenario, the learner receives both labeled and unlabeled data, but predictions are provided for the unlabeled only.

Learning Scenarios

- **On-line Learning** – in contrast with the previous scenarios, the online learning involves multiple rounds where training and testing are interleaved. At each round, the learner receives unlabeled data, makes predictions, receives the true label, and incurr a *loss*. The goal is to minimize the *cumulative loss* over all rounds.

- **Reinforcement Learning** – in this scenario the learner actively interacts with an *environment* – and in some cases affects it – and receives an *immediate reward* for each *action*. The goal is to maximize his reward over a course of actions and iterations with the environment, facing with the *exploration versus exploitation dilemma*.

- **Active Learning** – in this scenario the learner adaptively/interactively collects training examples by *querying an oracle*. The goal is to achieve a performance comparable to the standard supervised scenario (that is therefore "passive" learning), but with fewer labeled examples. Active learning is often used in applications where labels are expensive to obtain (e.g., computational biology applications).

- **Transfer Learning (Meta-Learning and Lifelong Learning)** – this scenario refers to advanced ML, with the learner learning continuously, accumulating the knowledge learned in the past, and using/adapting it to *help future learning and problem solving*. In the process, the learner becomes more and more knowledgeable and better and better at learning.

Learning tasks

- **Classification** – the aim is to assign a category (class), among a set of possible ones, to each example (e.g., image classification).
- **Regression** – the aim is to predict a real value to each example (e.g., prediction of stock values).
- **Ranking** – the aim is to learn how to order examples depending on some criterion (e.g., Web search).
- **Clustering** – the aim is to partitioning examples into *homogeneous* subsets (e.g., community detection in social networks).
- **Anomaly detection** – (aka **outlier detection**) is the identification of rare examples differing significantly from most of the data. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions (e.g., bank fraud or errors in a text).
- **Dimensionality reduction** or ***manifold* learning** – aimed at transforming an initial representation of examples into a lower-dimensional representation while preserving some properties of the initial representation.

Some notations and learning stages

- **Parameters** – as set of "coefficients" that the learning algorithm "fine-tunes" to fit the training data (e.g., the "thresholds" we used in the rules of our example).

- **Hyperparameters** – free parameters that are not determined by the learning algorithm, but rather specified as inputs to the learning algorithm to "drive" the tuning of the parameters.

- **Model** – the result of training: a model to make predictions on unseen data (i.e., the rules we have learned in our example).

Some notations and learning stages

- **Training data/set/sample** – examples used to train a ML algorithm (i.e., the table in our example). It is used for learning the values of the algorithm's parameters.

- **Validation data/set/sample** – examples used to validate the predictions capabilities provided by a trained model (we have not validation data in our example). It is used to learn the values of the hyperparameters.

- **Test data/set/sample** – examples used to test the predictions capabilities provided by a final ML model (we have not validation data in our example). Neither parameters nor hyperparameters are changed during this stage.

About "experience"…

- Every learner must deal with the **induction problem**. Bertrand Russel liked to illustrate the problem with the story of the **inductivist turkey**.*

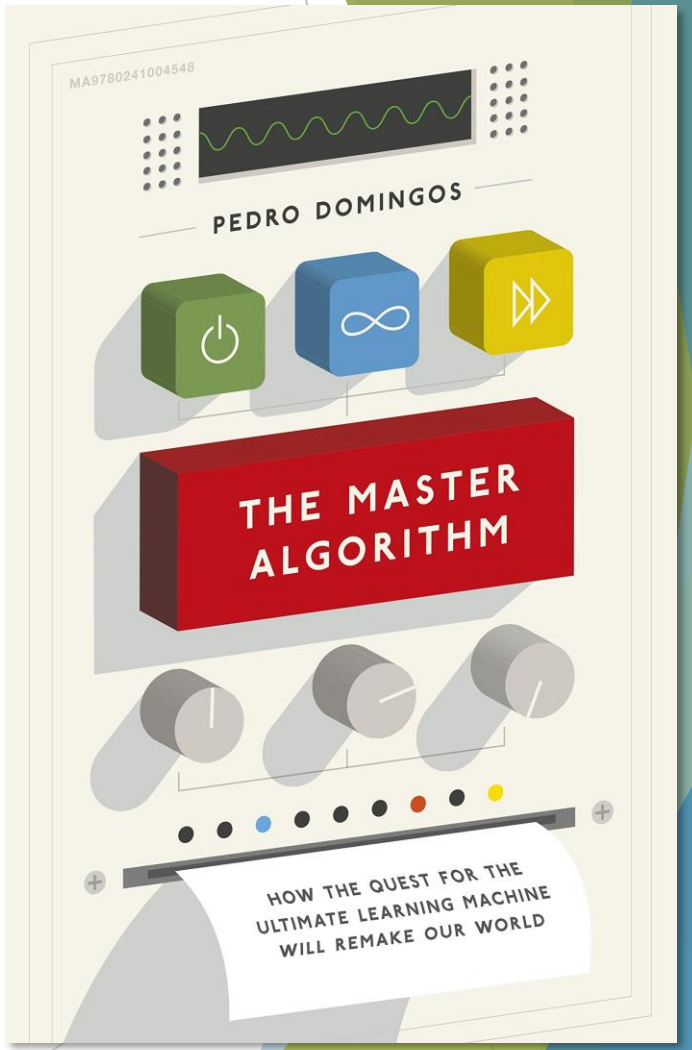
- On his first morning at the farm, the turkey was feed ar 09:00 a.m.
- Being a good inductivist, he collected many observations before jumping to conclusions.
- Having been fed consistenly at 09:00 a.m. for many consecutive days, he finally concluded that he would always be fed at 09:00 a.m.
- Then came the morning of Christmas eve, and his throat was cut.

** Pedro Domingos "The Master Algorithm"*

The Master Algorithm

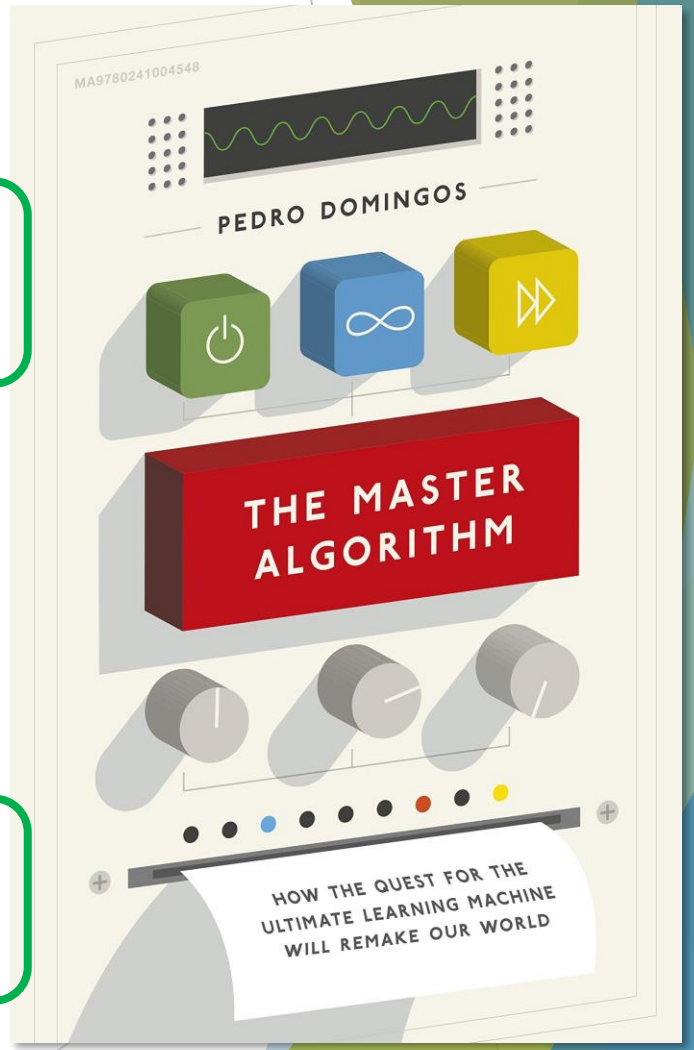
- The central hypothesis of the Pedro Domingo's book:

All knowledge – past, present, and future – can be derived from data by a single, universal learning algorithm



The five tribes of ML

- **Symbolists** – they have figured out how to incorporate preexisting knowledge into learning, and how to combine different pieces of knowledge to solve new problems (i.e., inverse deduction).
- **Connectionists** – learning is what the brain does (artificial neural networks).
- **Evolutionaries** – mother of all learning is natural selection (genetic programming: from weights tuning to structure learning).
- **Bayesians** – are concerned above all with uncertainty. All learned knowledge is uncertain, and learning itself is a form of uncertain inference (probabilistic inference).
- **Analogizers** – the key to learning is recognizing similarities between situations and thereby inferring other similarities (support vector machines and kernel learning methods.



Machine Learning for forecasting

- The task of interest is **forecasting future values of a given time series**, and the experience is constituted by the **past values** of such time series and **other statistics**.

- The models progressively tune their parameters by a two-step process wherein the data is divided into a **training set**, where the learning from experience process takes place, and a **validation set**, where the performance of the model is evaluated with respect to **some performance measure** (e.g., minimizing residual sum of squares) and **succeeding adjustments of the model are made**.

- In the case of time series usually the training and the validation sets are taken as **two consecutive time periods of the series** of varying length.

- other approaches: **cross-fold validation** to "fully" exploit the available set of data.

Performance measures for time series

Mean Squared Error

$$MSE = \frac{1}{m}\sum_{t=1}^{m}(y_t - \hat{y}_t)^2$$

$$MAE = \frac{1}{m}\sum_{t=1}^{m}|y_t - \hat{y}_t|$$

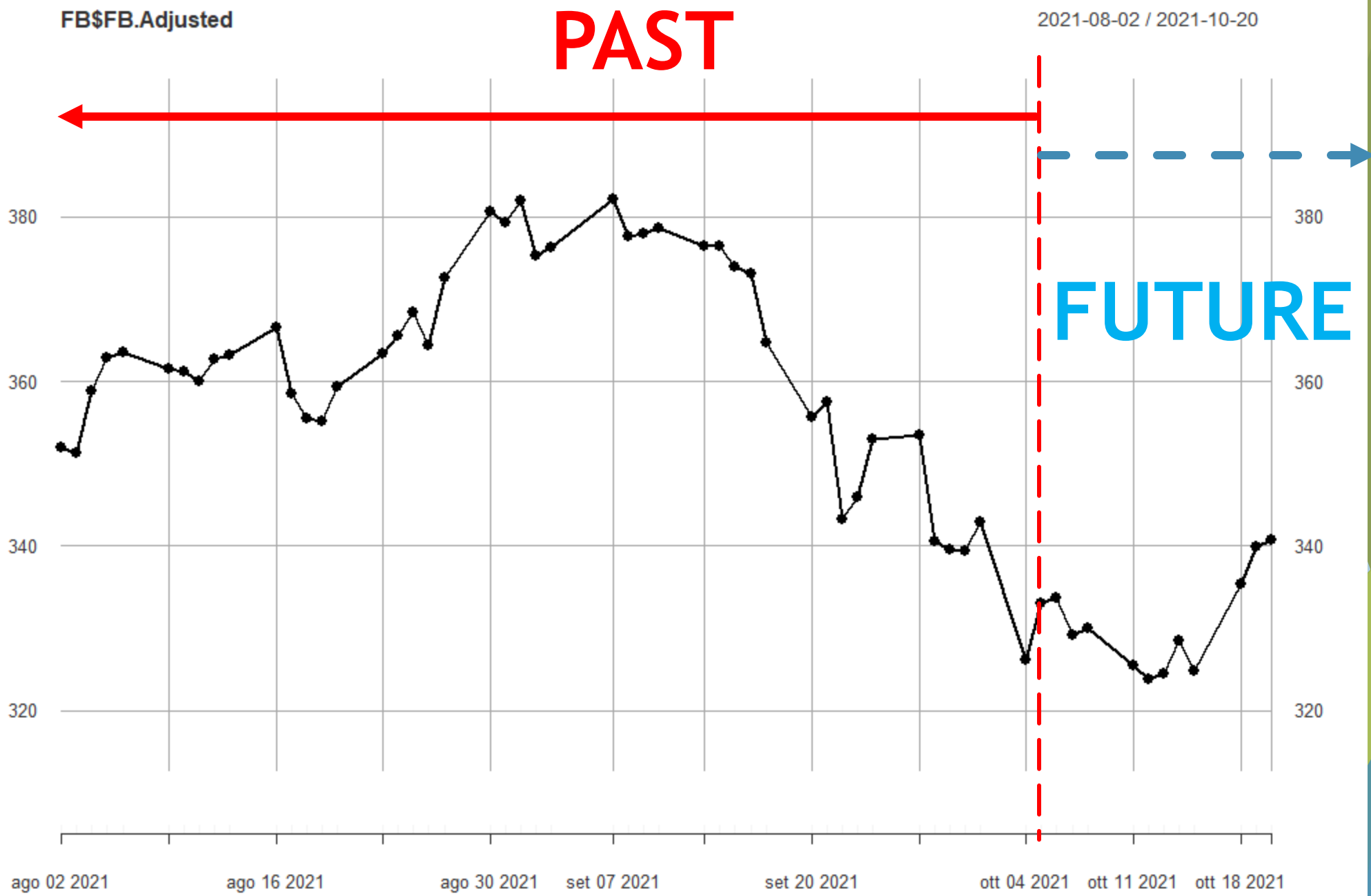
Mean Absolute Error (or MAD, Mean Absolute Deviance)

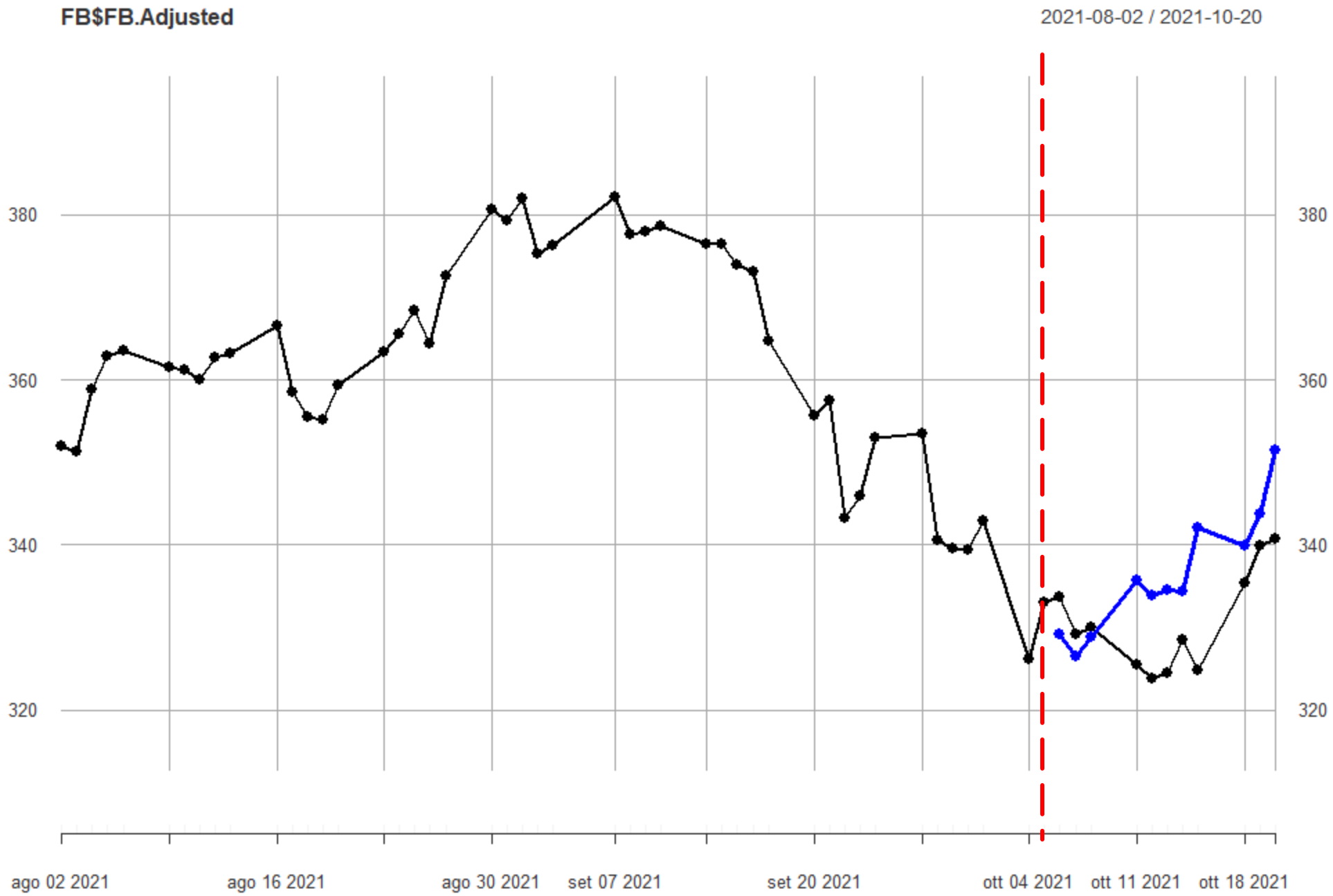
$$RMSE = \sqrt{\frac{1}{m}\sum_{t=1}^{m}(y_t - \hat{y}_t)^2} = \sqrt{MSE}$$

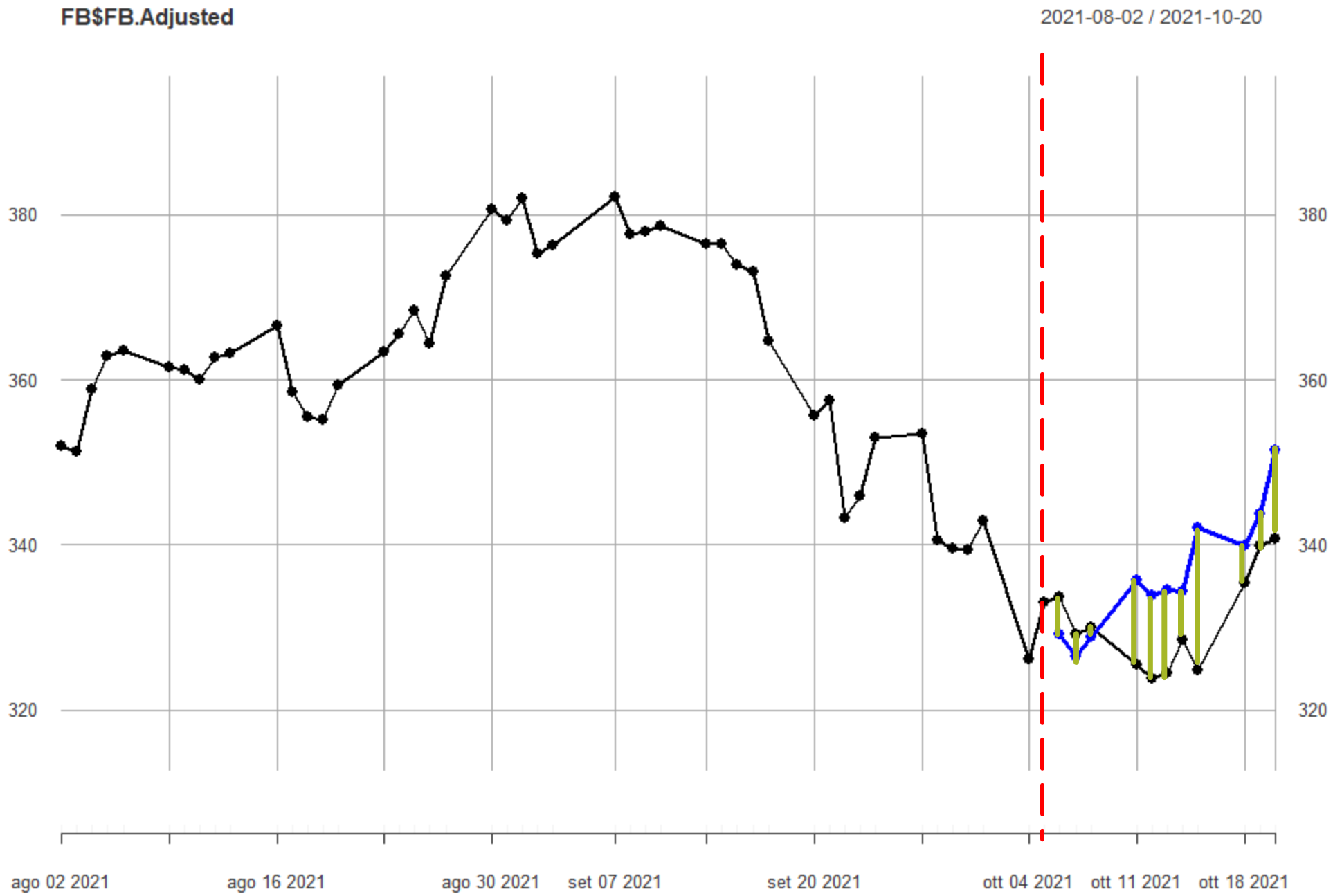
Root Mean Squared Error

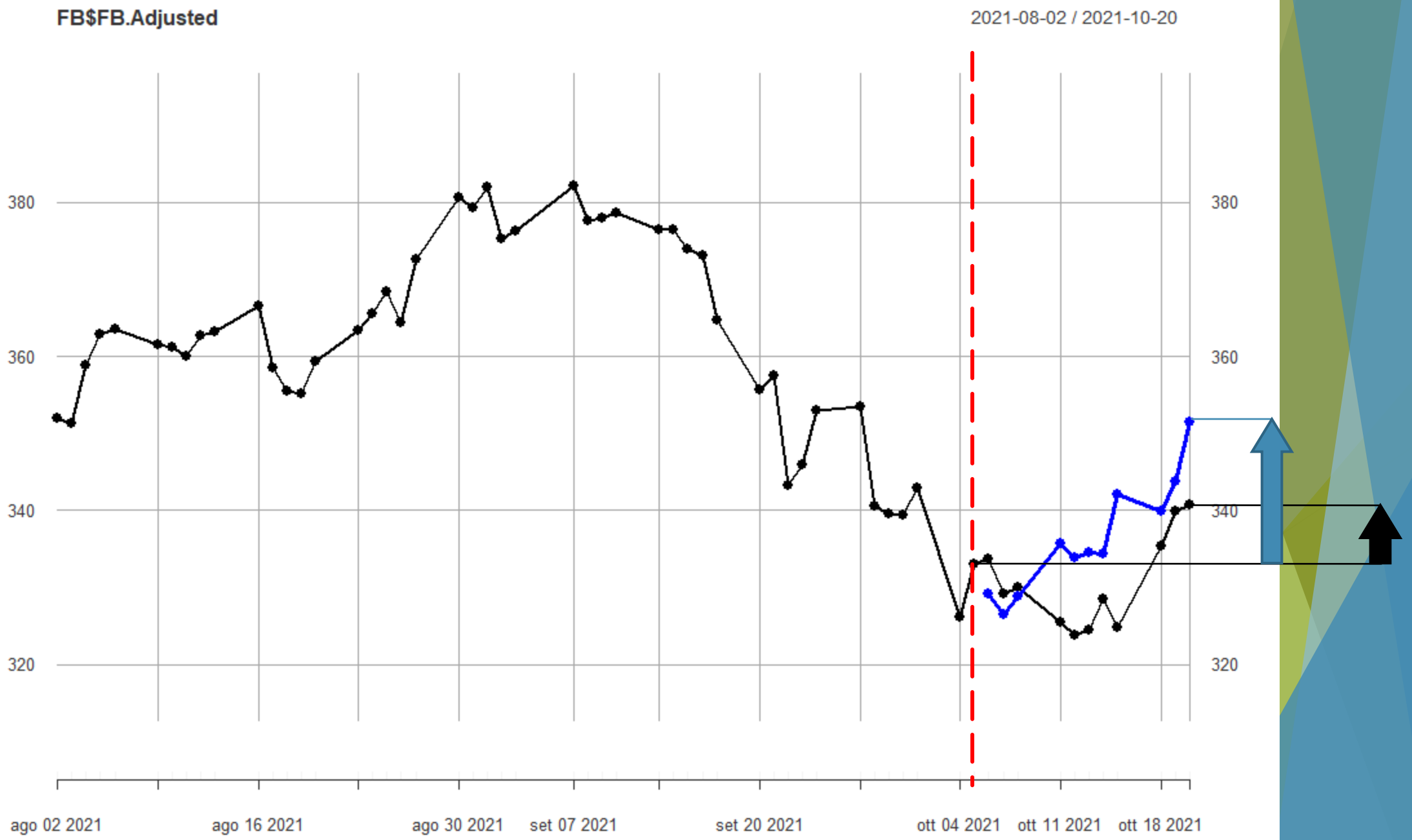
$$MAPE = \frac{1}{N}\sum_{k=1}^{N}\frac{|F_k - A_k|}{A_k}$$

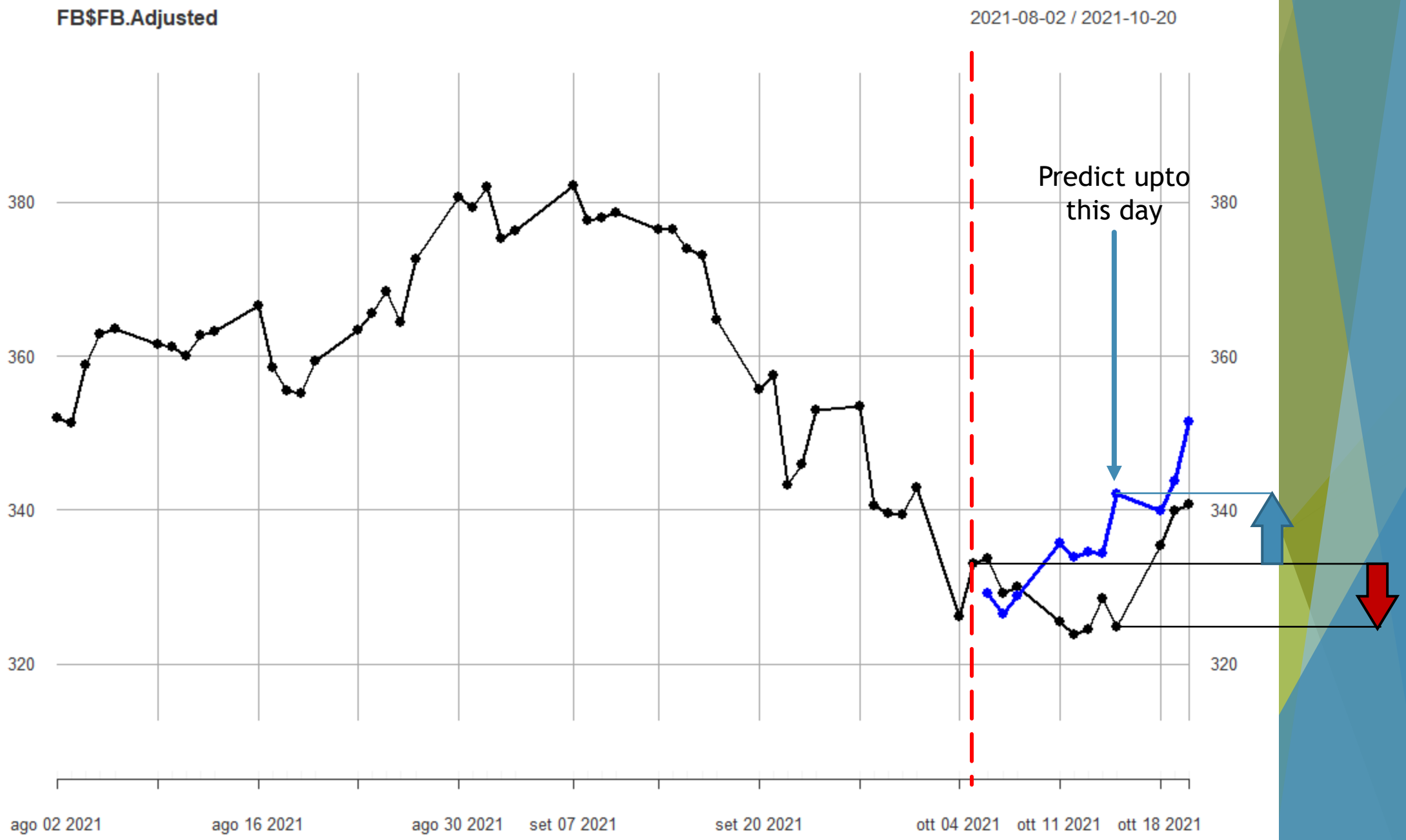
Mean Absolute Percentage Error (F is «forecast», A is «actual»)

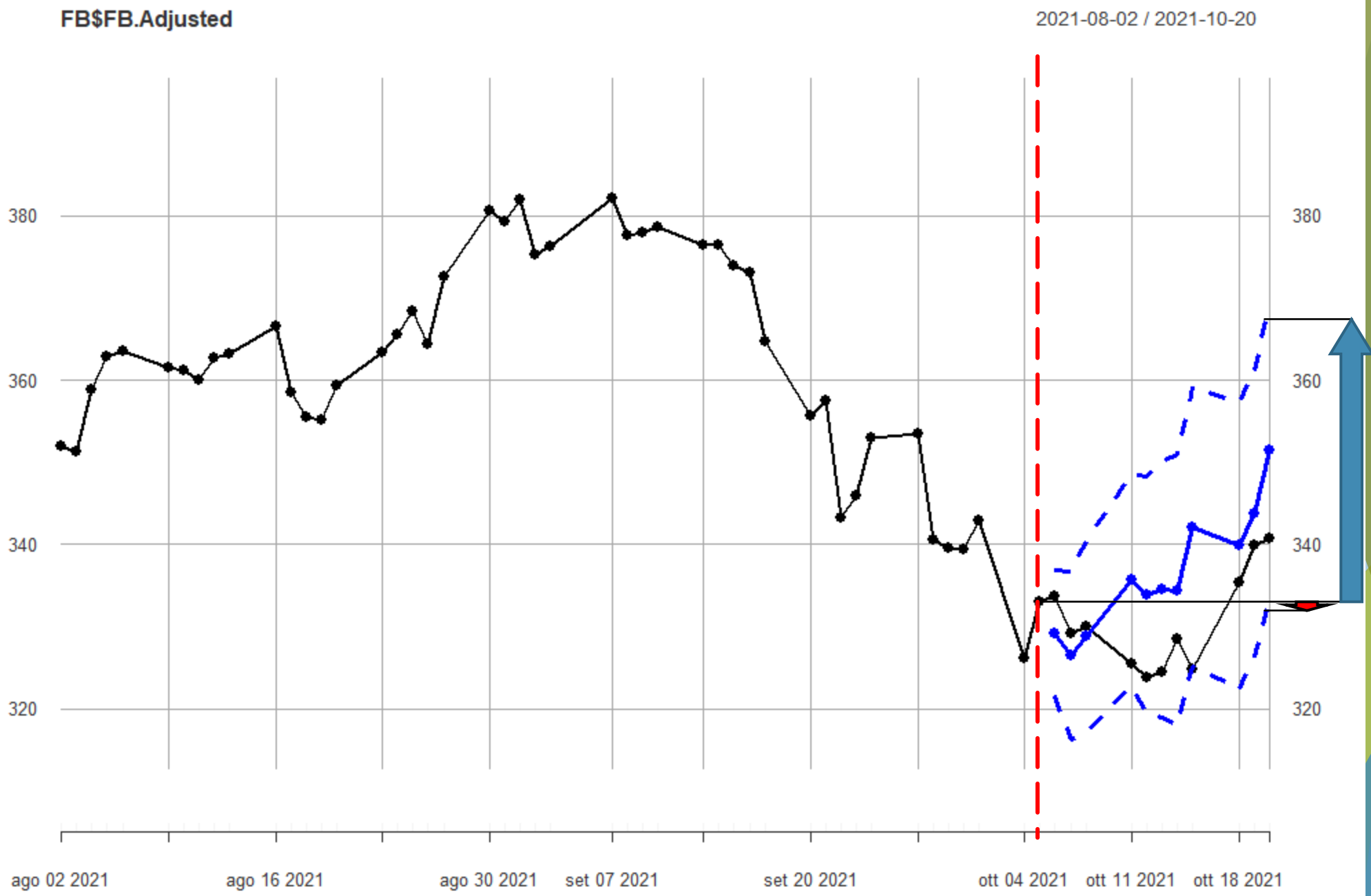

FB$FB.Adjusted
PAST
2021-08-02 / 2021-10-20
FUTURE
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021

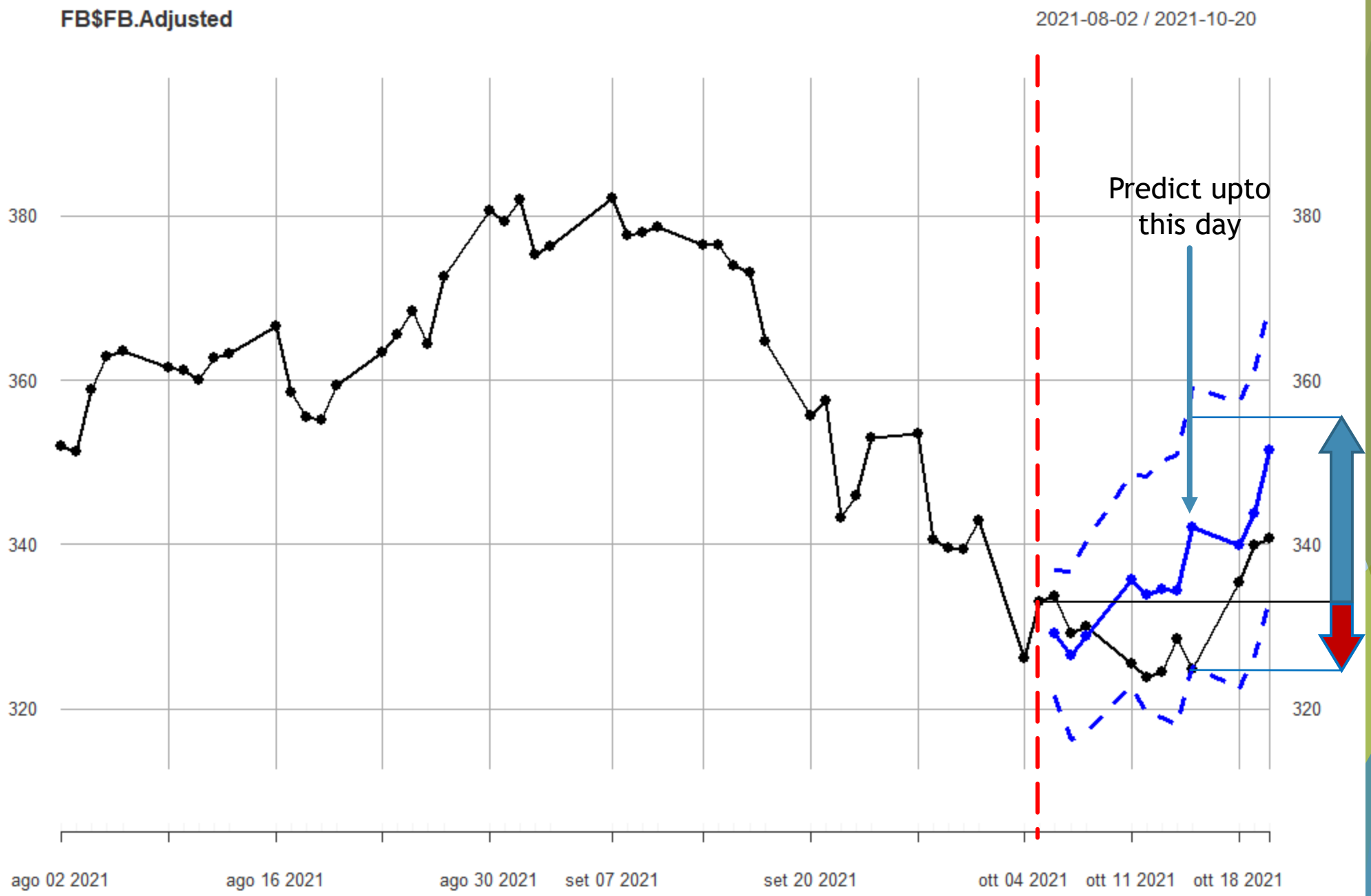

FB$FB.Adjusted
2021-08-02 / 2021-10-20
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021


FB$FB.Adjusted
2021-08-02 / 2021-10-20
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021


FB$FB.Adjusted
2021-08-02 / 2021-10-20
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021


FB$FB.Adjusted
2021-08-02 / 2021-10-20
Predict upto this day
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021


FB$FB.Adjusted
2021-08-02 / 2021-10-20
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021


FB$FB.Adjusted
2021-08-02 / 2021-10-20
Predict upto this day
380
360
340
320
ago 02 2021
ago 16 2021
ago 30 2021
set 07 2021
set 20 2021
ott 04 2021
ott 11 2021
ott 18 2021